Docket No.: 31862.000065

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Baller, Eric Henry, et al.	Examiner:	Vaughn Jr., William C
Serial No.	10/028,126	Art Unit:	2143
Filed:	12/20/01		

Title: METHOD AND APPARATUS FOR MANAGING INTELLIGENT ASSETS IN A DISTRIBUTED ENVIRONMENT

DECLARATION UNDER 37 CFR 1.131

We, Eric Henry Baller, John Thomas Canosa, John Maitland Cook III, David Patrick Hart, Christopher James Kuntz, John Louis Taylor, Dimitrios Psarros, Thomas Chiarella, and Rajeev Raman hereby declare that:

1. We are named as joint inventors for the invention disclosed and claimed in US Patent Application No. 10/028,126 filed on December 20, 2001 and hereinafter referred to as the "subject application".

2. All of the pending claims 1-31 in the subject application stand rejected under 35 USC 103(a) as being unpatentable over US Patent No. 6,430,711 to Sekizawa in view of US Application Publication No. 2003/0072027 corresponding to US Patent Application No. 09/976,625 of Haines et al., the latter hereinafter referred to as the "Haines application".

3. The Haines application has an effective date as a reference corresponding to its US filing date of October 11, 2001, which is a little more than two months before the filing date of the subject application.

4. Before October 11, 2001, the effective date of the Haines application, the invention set forth in claims 1-31 of the subject application was completed in this country in the form of an actual reduction to practice.

5. Evidence of the completion of the claimed invention is found in attached Exhibits A-E, all of which were prepared by Questra Corporation, the assignee of record, as internal documentation before October 11, 2001, the effective date of the Haines application.

6. Exhibit A includes relevant portions of a document entitled "A2B.Platform Architecture", which describes the A2B Platform an assembly of components for connecting remote devices to a central system. Figure 1 illustrates the conceptual

domain of the A2B.Platform. Figure 2, which appears at the bottom of page 2, illustrates an example of an A2B appliance-based community, showing an A2B Device Framework for adapting the devices for Internet communication with and A2B Enterprise Framework that further communicates with an enterprise system. Figure 4 illustrates a SOAP message structure for communicating with remote devices corresponding to Figure 4 of the subject application.

7. Exhibit B includes relevant portions of a document entitled "A2B Technology & Packaging", which illustrates three connection options including Option 1: No bridge necessary, Option 2: A2B Device-side Bridge, and Option 3: A2B Server-side Bridge.

8. Exhibit C includes relevant portions of a document entitled "Remote Control Management Module Design Document" describing the framework for connecting remote machines to a central system in accordance with the invention.

a) The "Remote Control Management Module Design Document" was prepared before the effective date of the Haines application for the benefit of those "who maintain update and support the implementation of the module" (see page 1 under the heading "Audience"). It is also instructive to note that under the heading "Implementation Issues" on the same page is the indication of "none", evidencing no impediments to the implementation of the module.

b) A component diagram of the remote control management system for connecting a server to a remote device is shown in Figure 1 on page 2. The diagram shows the distribution of components of the remote control management system between a web server, and A2B "Enterprise" server, and a remote device.

c) A diagram depicting a database design that supports the information requirements of the remote control management system appears in Figure 2 on page 3.

d) An interface used by an application (web server) with the A2B server is depicted and described with reference to Figure 12 as a remote control management proxy.

e) An interface used by the A2B server with the remote device server is depicted and described with reference to Figure 13 as a remote control proxy.

f) Table 6 on page 21 shows the configuration of SOAP encoded messages enabling remote devices to register its web service. Encoded within the messages are the "MemberId" and "Password" of the device.

g) Table 12 spanning pages 24 and 25 shows the configuration of a SOAP encoded messages used to query and respond concerning web service profiles, property and command metadata and a SOAP encoded response from a remote device. Note that a "MemberId" and "Password" are embedded within the message.

h) Table 16 on pages 27 and 28 documents a SOAP message and response for transmitting a list of property names and for returning with a list of values for the requested properties.

i) Table 19 on pages 29 and 30 documents a SOAP message for setting a list of properties on remote devices.

9. Exhibit D is a document entitled "A2B Messaging Context Diagram" illustrating communications across firewalls.

10. Exhibit E is a spreadsheet document evidencing the completion of the various interface and framework components of the invention through version v2.1, all before October 11, 2001, the effective date of the Haines application.

11. We declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false
validity of the application or any patent issued thereon.

Signatures of Declarants:

Eric Henry Baller	9/13/04 Date
John Thomas Canosa	9/13/04 Date
John Maitland Cook III	Date
David Patrick Hart	9/13/04 Date

John Louis Taylor — 9/13/2004
Date

Dimitrios Psarros — 09/13/2004
Date

Thomas Chiarella
Date

Rajeev Raman — 09/13/2004
Date

EXHIBIT A

QUESTRA

A2B.Platform Architecture

BUSINESS CONFIDENTIAL

[redacted]

© Questra Corporation
350 Linden Oaks
Rochester, New York 14625
Phone: 716.381.0260
Fax: 716.381.8098

1 Introduction

The Appliance-Based Community leverages the infrastructure of the Internet to enable not only people to device collaboration, but electronic transactions and additional services surrounding these transactions. At the heart of the Appliance-Based Community, the A2B.Platform provides the on-line community A2B Web Services that bring together people and devices to leverage the *network effect* of the Internet. The Questra A2B.Plaform is based on an architecture that will enable new or existing Internet communities or enterprise systems to enhance competitive advantage by becoming "A2B™-enabled". The following *Figure 1 – Appliance-Based Community Conceptual Architecture* provides a graphical depiction of the conceptual domain in which the A2B.Platform may exist.

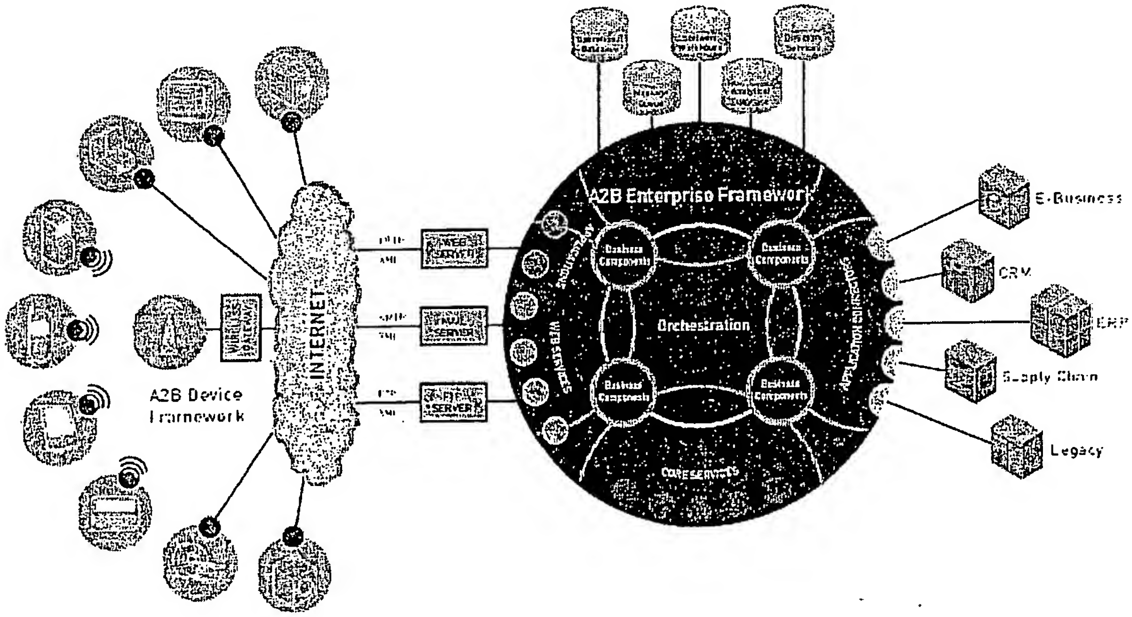


Figure 1 – A2B.Platform Conceptual Architecture

1.1 Purpose

This *A2B.Platform Architecture* document articulates the following aspects of the A2B.Platform architecture:

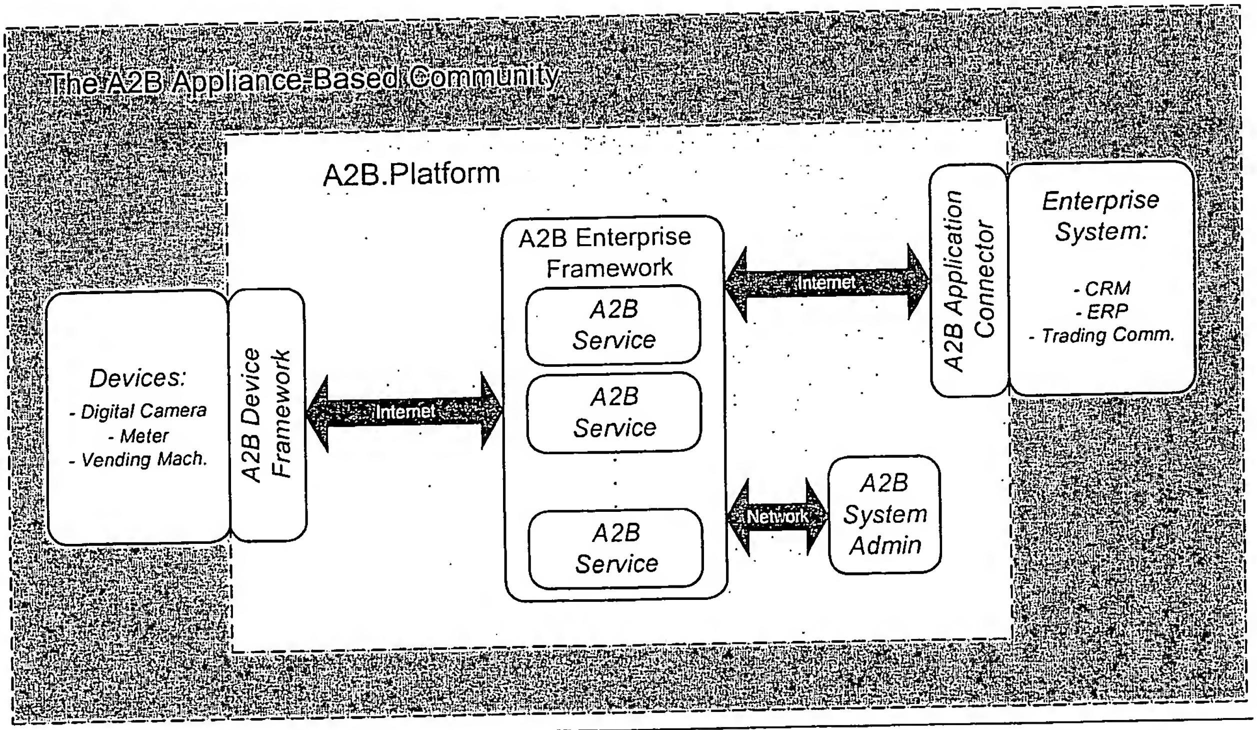
- **Architecture Principles:** specifies those guiding principles that frame the definition of an architectural model for enabling A2B integration
- **Architecture Requirements:** defines the specific A2B.Platform system-level requirements for a robust, open, reliable and scalable A2B.Platform architecture

- **A2B Layered Architecture:** describes the protocol stack leveraged by the A2B.Platform for cross-language, cross platform Appliance-Based Community integration
- **A2B Service Description Language:** specifies the interface and invocation methods comprising the operational information for an A2B Service
- **a2bML Language:** defines the structure of the information that will be exchanged in the invocation of an A2B Web Service
- **A2B Messaging Model:** describes the lightweight protocol that provides a mechanism for exchanging information in a distributed environment using XML
- **A2B Service Discovery:** specifies the method to publish and discover information about A2B Services

This document will remain evergreen throughout the evolution of the A2B.Platform Architecture, as Questra intellectual property is harvested from successive A2B customer project implementation baselines. Furthermore, this document will be leveraged as an architecture foundation upon entry into the Strategy and Innovation Phases of new Questra client A2B engagements.

1.2 Scope

The A2B.Platform portion of an Appliance-Based Community includes the A2B Device Framework, the A2B Enterprise Framework, the Web Services, and any A2B Application Connector used to connect to an enterprise application. This context is depicted in the following *Figure 2 – A2B.Platform Context Diagram*. Refer to *A2B.Platform – System Requirements Specification* Section 2.1 for detailed discussion on the scope of the A2B.Platform.



2.5 The a2bML Language

The a2bML language will define the structure of the information that will be exchanged in the invocation of an A2B Service. A2B Services will be able to share common structures and also define their own set of message requirements. All definitions will be part of the a2bML language. This section will include detailed descriptions of the supported message formats.

In addition to providing for structured information exchange, the a2bML language defines the following aspects of the programming interfaces for invoking A2B Services:

- Interface signatures for A2B Services
- Endpoints where A2B Services can be contacted

2.6 A2B Messaging Model

A2B Services will operate on SOAP messages that will hold a2bML data. SOAP is a lightweight protocol that provides a mechanism for exchanging information in a distributed environment using XML. A SOAP message consist of three parts:

- The SOAP envelop that provides the framework for packaging message information
- The SOAP encoding rules that defines the mechanism for data serialization
- The SOAP RPC representation that defines the convention to represent remote procedure calls and responses.

The SOAP specification defines also bindings to transport SOAP messages using the HTTP protocol.

The following diagram depicts the structure of a SOAP message:

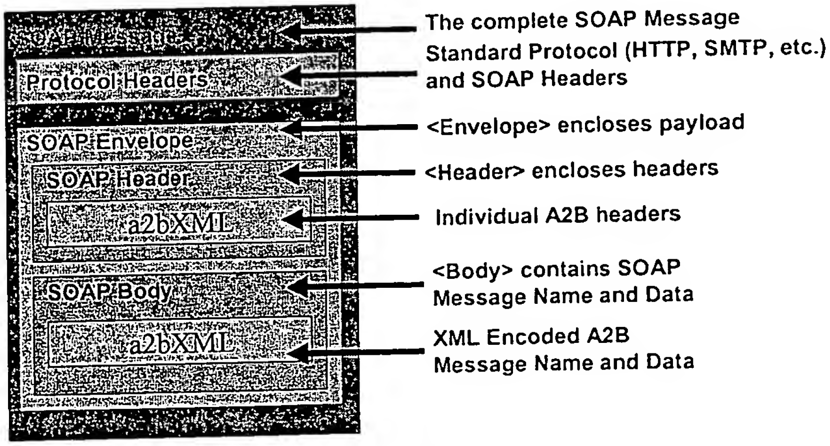

SOAP Message
Protocol Headers
SOAP Envelope
SOAP Header
a2bXML
SOAP Body
a2bXML
The complete SOAP Message
Standard Protocol (HTTP, SMTP, etc.) and SOAP Headers
<Envelope> encloses payload
<Header> encloses headers
Individual A2B headers
<Body> contains SOAP Message Name and Data
XML Encoded A2B Message Name and Data

Figure 4 - SOAP Message Structure

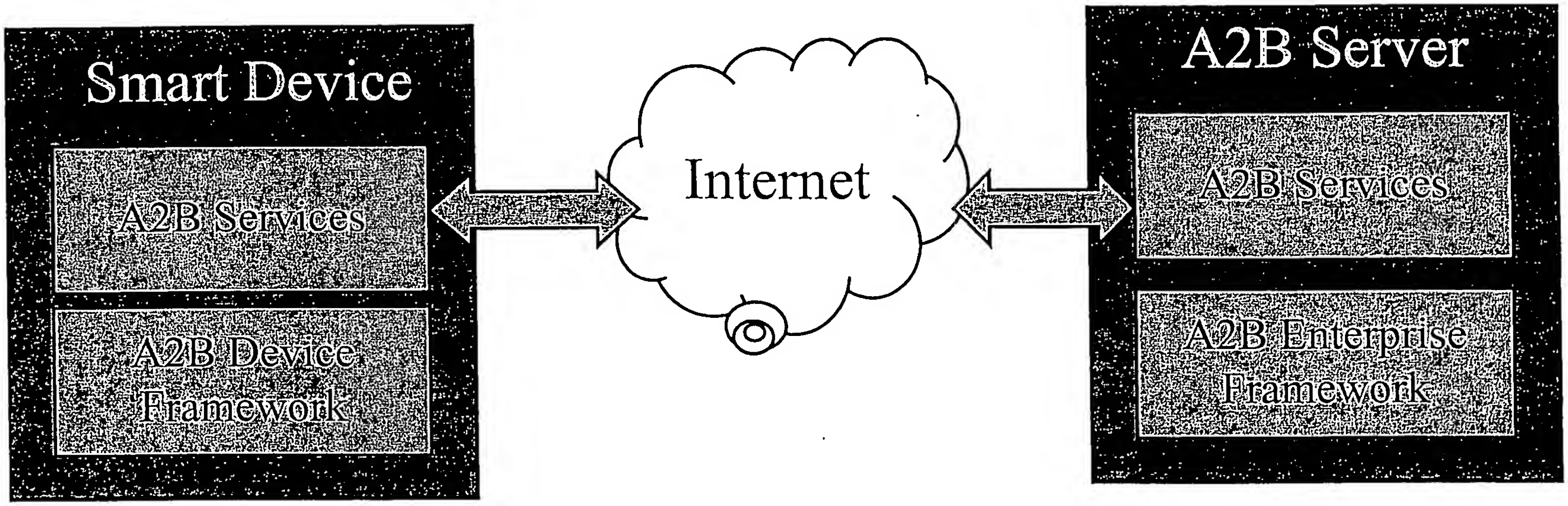
EXHIBIT B

A2B Technology & Packaging

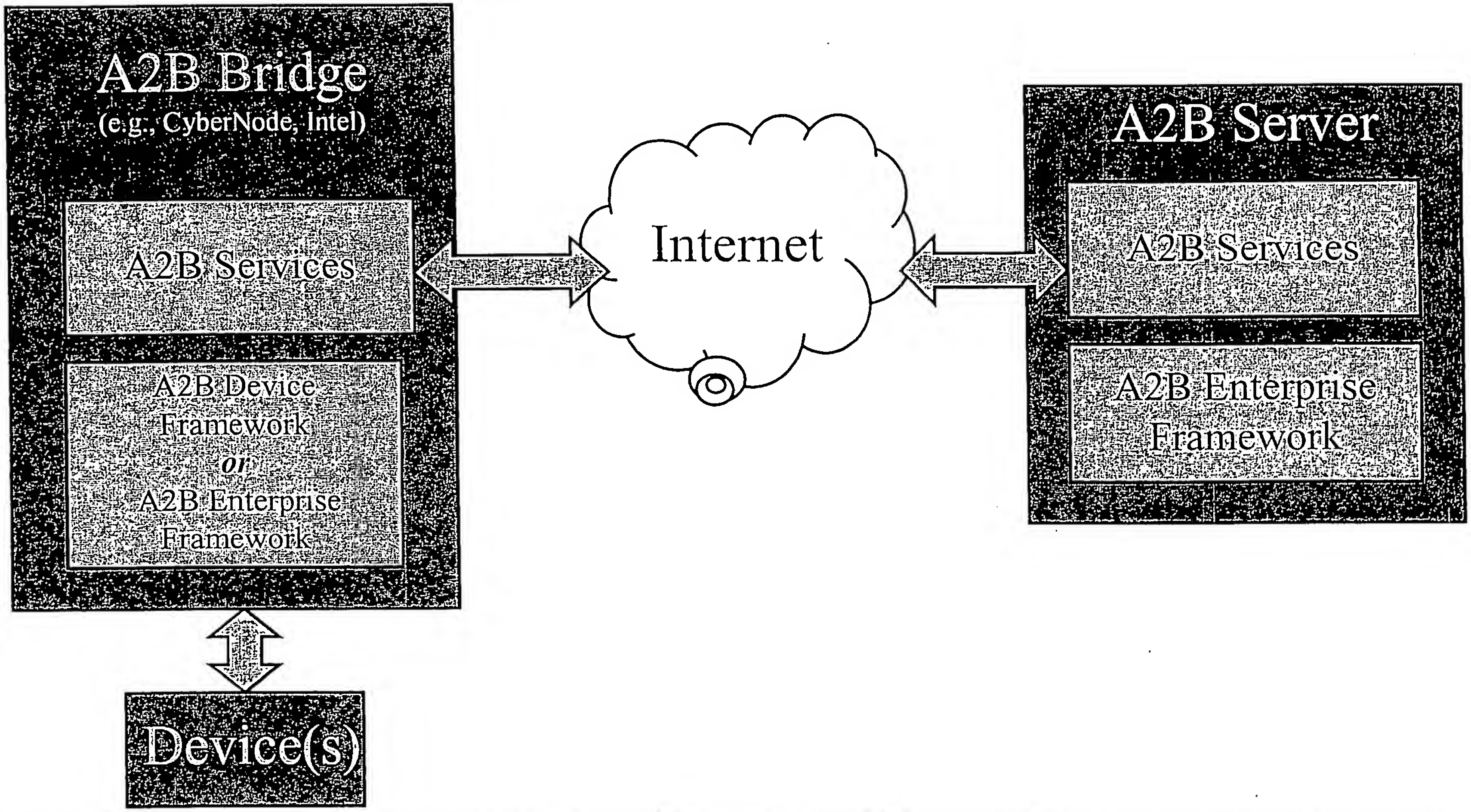
A2B Solution Architecture Team

[redacted]

Connecting Devices
Option 1: No bridge necessary



Connecting Devices
Option 2: A2B Device-side Bridge



Connecting Devices
Option 3: A2B Server-side Bridge

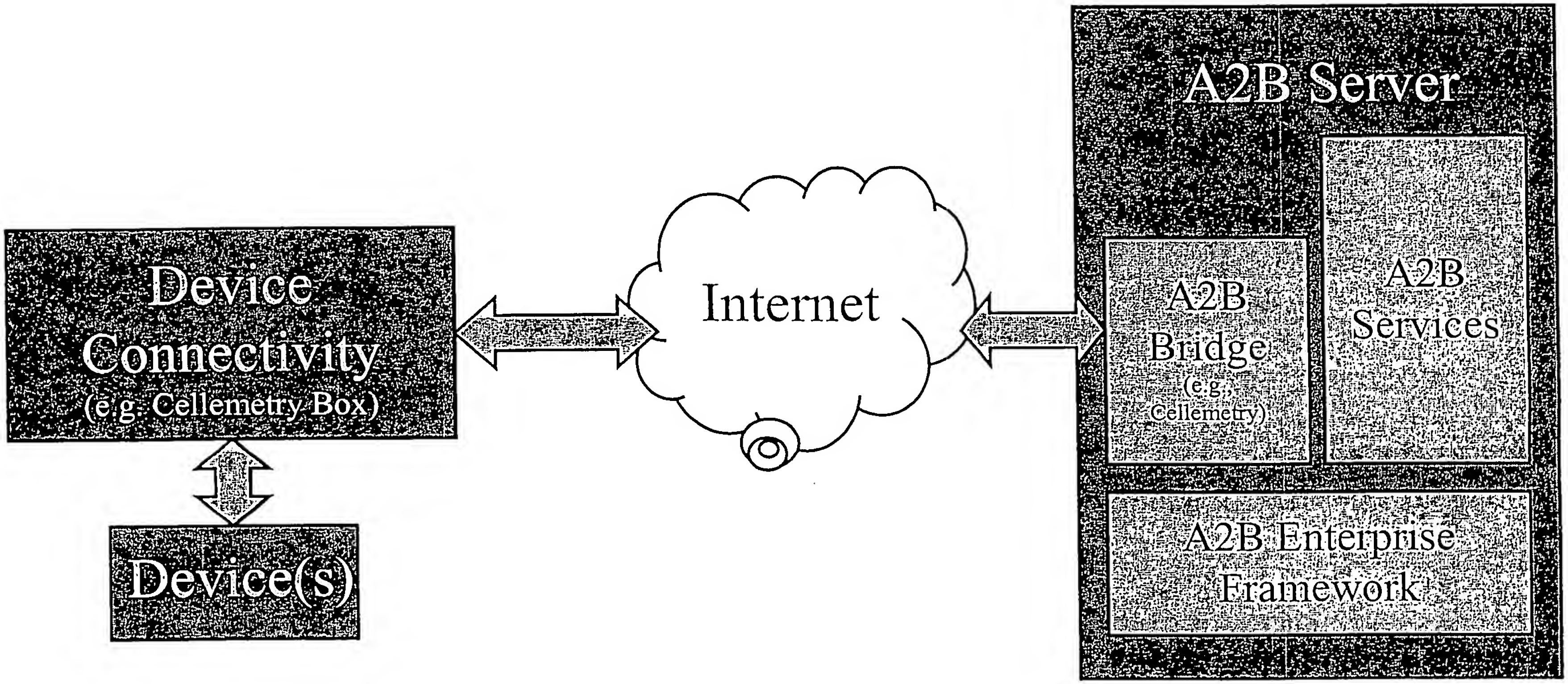


EXHIBIT C

QUESTRA

REMOTE CONTROL MANAGEMENT MODULE DESIGN DOCUMENT

VERSION 2.1

PROPRIETARY INTERNAL DOCUMENTATION
NOT FOR EXTERNAL DISTRIBUTION

[redacted]

© Questra Corporation
350 Linden Oaks
Rochester, New York 14625
Phone: 716.381.0260
Fax: 716.381.8098

THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF Questra Corporation, AND MAY NOT BE USED OR COPIED IN WHOLE OR IN PART WITHOUT THE EXPRESS WRITTEN CONSENT OF Questra Corporation.

1 Purpose

This document describes the detailed implementation design of the *Remote Control Management* module of the A2B product.

2 Audience

The intended audiences of this document are architects, designers, developers, database designers, database administrators, and quality assurance teams, who maintain update and support the implementation of the module.

3 Requirements

There are no formal requirements documents for the Remote Control Management Module.

4 Scope

TDB

5 Implementation Issues

None.

6 Overview

The Remote Control Management Module consists of two web services:

- Remote Control Management Web Service
- Remote Control Web Service

The Remote Control Management web service is hosted on an enterprise server and handles remote control requests from the management console web application or other applications. When a request is made, the Remote Control Management web service resolves where the specified device is located and passes the request on to the Remote Control web service on that device.

The Remote Control web service is hosted on a remote device and is responsible for handling requests from the Remote Control Management web service. The Remote Control web service interfaces with the device API to perform the requested operation, then returns the results to the Remote Control Management web service.

The following is a component diagram of the Remote Control Management web service.

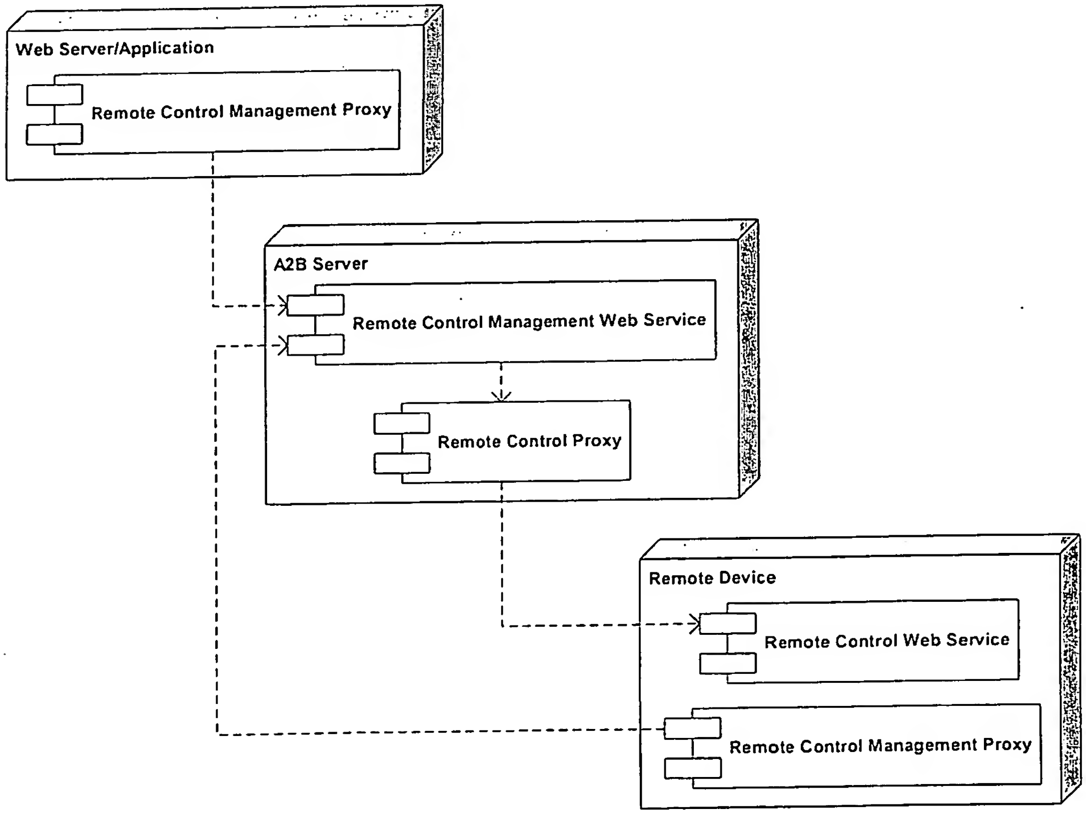

Web Server/Application
Remote Control Management Proxy
A2B Server
Remote Control Management Web Service
Remote Control Proxy
Remote Device
Remote Control Web Service
Remote Control Management Proxy

Figure 1 Remote Control Management Component Diagram

7 Remote Control Management – Enterprise Framework

7.1 Information Architecture

The following diagram depicts the database design that supports the information requirements of the Remote Control Management module.

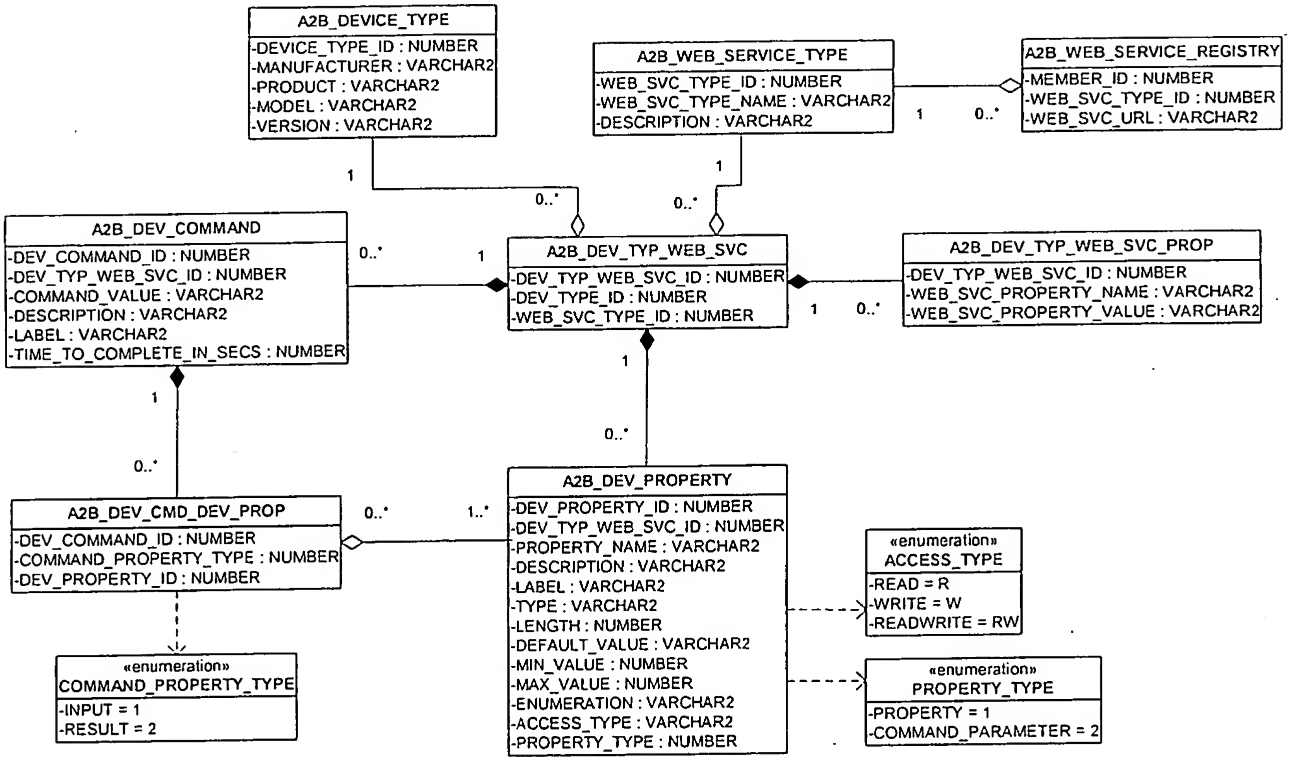


Figure 2 Database design that supports the Remote Control Management module.

Table	Description
A2B_WEB_SERVICE_REGISTRY	Holds the web service URL for devices which have registered their web service.
A2B_DEVICE_TYPE	Holds the device types.
A2B_WEB_SERVICE_TYPE	Holds the web service types.
A2B_DEV_TYP_WEB_SVC	Associates a device type and web service type.

7.2.3 Remote Control Management Proxy

The *RemoteControlManagementProxy* class is used by an application to interface with the Remote Control Management web service on the enterprise server.

The main classes in the Remote Control Management Proxy are shown in the following class diagram.

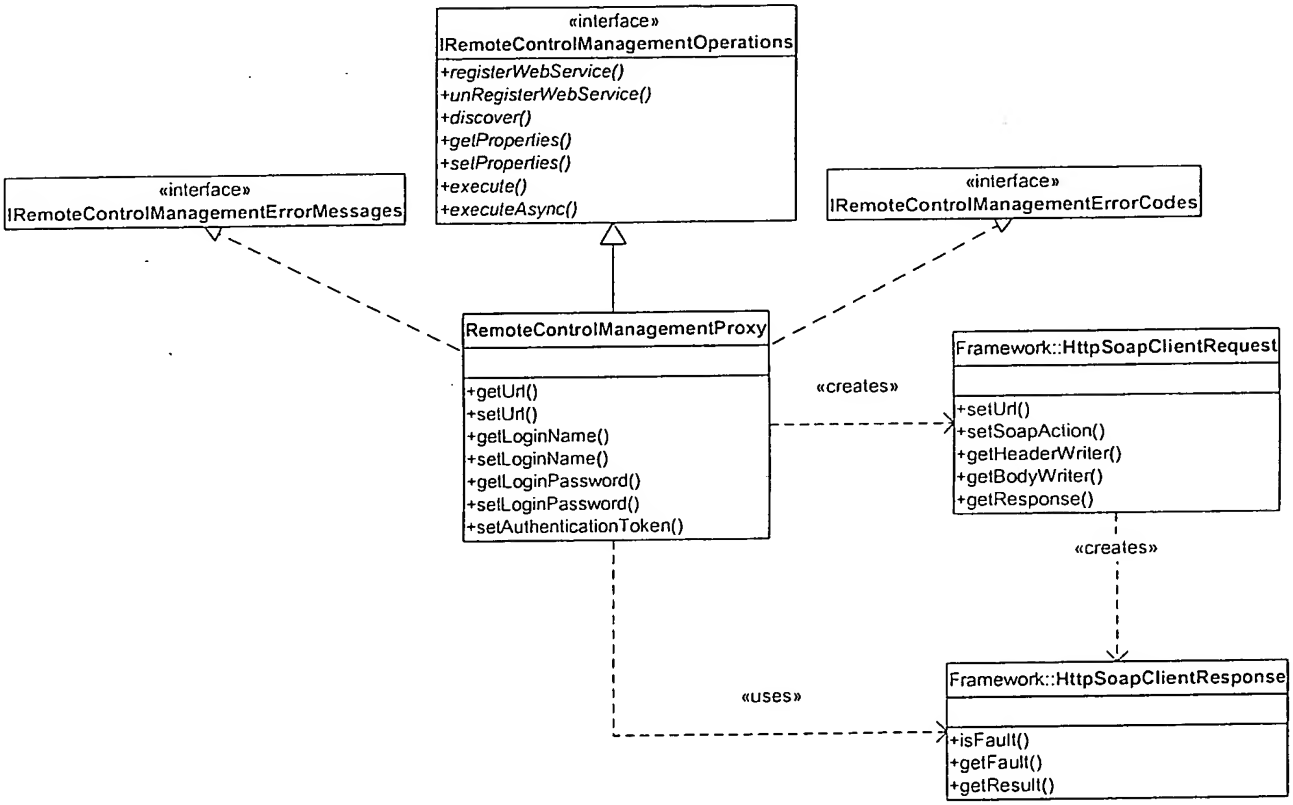


Figure 12 Remote Control Management Proxy Class Diagram

The *IRemoteControlManagementOperations* interface defines the methods that the web service supports, while the *RemoteControlManagementProxy* class provides an implementation of that interface. The *IRemoteControlManagementErrorMessages* and *IRemoteControlManagementErrorCodes* contain the text error messages and error codes encapsulated in an exception.

The *RemoteControlManagementProxy* class uses the *HttpSoapClientRequest* class to serialize the request to the Remote Control Management web service located on the enterprise server, and uses the *HttpSoapClientResponse* class to unserialize the response from the web service.

7.2.4 Remote Control Proxy

The *RemoteControlProxy* class is used by the Remote Control Management web service to interface with the Remote Control web service located on a remote device.

The main classes in the Remote Control Proxy are shown in the following class diagram.

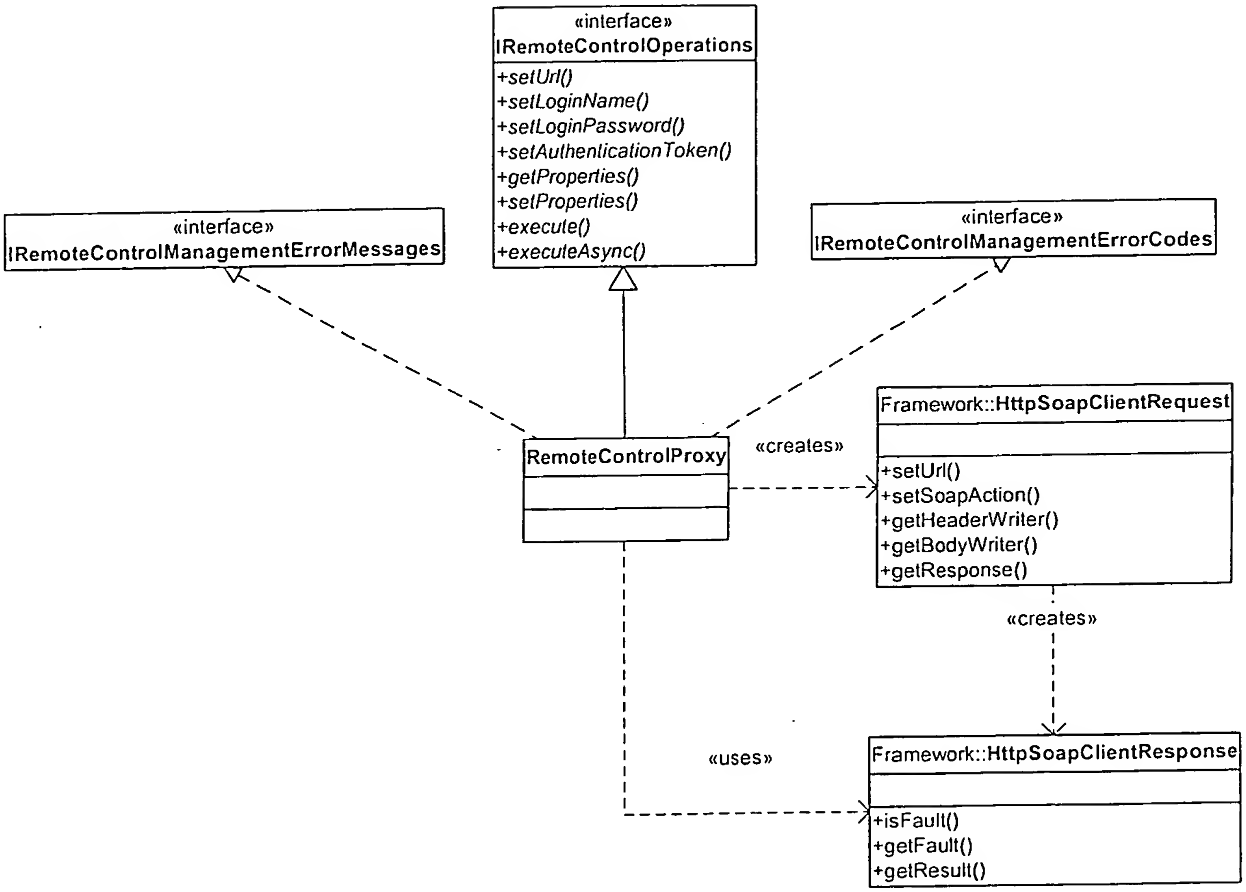


Figure 13 Remote Control Proxy Class Diagram

The *IRemoteControOperations* interface defines the methods that the web service supports, while the *RemoteControlProxy* class provides an implementation of that interface. The *IRemoteControlManagementErrorMessages* and *IRemoteControlManagementErrorCodes* contain the text error messages and error codes encapsulated in an exception.

The *RemoteControlProxy* class uses the *HttpSoapClientRequest* class to serialize the request to the Remote Control web service located on a remote device, and uses the *HttpSoapClientResponse* class to unserialize the response from the web service.

Table 5 Remote Control Management web service operations

7.5.1.1 registerWebService Operation

This method enables a remote device to register its web service with the Web Service Registry.

Web Service: Remote Control Management	**Web Method:** registerWebService
SOAP Action: http://www.questra.com/a2b/RemoteControlManagement#registerWebService	

SOAP Request Message:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance/">
  <SOAP-ENV:Header>
    <RequestHeader>
      <Operation>registerWebService</Operation>
      <ServiceName>RemoteControlManagement</ServiceName>
      <SecurityContext>
        <Credentials>
          <MemberId>a2bdevice</MemberId>
          <Password>a2bdevice</Password>
        </Credentials>
      </SecurityContext>
    </RequestHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <registerWebService>
      <MemberId>a2bdevice</MemberId >
      <WebServiceType>RemoteDiagnostics</WebServiceType>
      <WebServiceURL>
        http://localhost:8081/a2b/RemoteDiagnostics
      </WebServiceURL>
    </registerWebService>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Response Message:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/"
xmlns:xsi=http://www.w3.org/1999/XMLSchema/instance/>
  <SOAP-ENV:Header>
    <ResponseHeader version="1.0"></ResponseHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <registerWebServiceResponse></registerWebServiceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Table 6 registerWebService web method sample messages.

	• RemoteDiagnostics • RemoteControl

Table 10 unRegisterWebService web method parameters.

The method can raise a *RemoteControlServiceExcepton* exception with the following codes:

Exception Code
REMOTE_CONTROL_MANAGEMENT_GENERAL_ERROR
NOT_AUTHORIZED
INVALID_PARAMETER
INVALID_CONTENT

Table 11 unRegisterWebService web method exception codes.

7.5.1.3 discover Operation

This method is used by an application to query for web service profiles and property and command metadata.

Web Service: Remote Control Management	**Web Method:** discover
SOAP Action: http://www.questra.com/a2b/RemoteControlManagement#discover	

SOAP Request Message:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance/">
  <SOAP-ENV:Header>
    <RequestHeader>
      <Operation>discover</Operation>
      <ServiceName>RemoteControlManagement</ServiceName>
      <SecurityContext>
        <Credentials>
          <MemberId>a2btest</MemberId>
          <Password>a2btest</Password>
        </Credentials>
      </SecurityContext>
    </RequestHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <discover>
      <RequestType>DISCOVER_WEB_SERVICES</RequestType>
      <Restrictions>
        <Restriction name="WebServiceType">RemoteDiagnostics</Restriction>
      </Restrictions>
      <Properties>
        <Property name="MemberId">a2bdevice</Property>
      </Properties>
    </discover>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

[redacted]

SOAP Response Message:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/"
xmlns:xsi=http://www.w3.org/1999/XMLSchema/instance/>
  <SOAP-ENV:Header>
    <Copyright>
      Copyright (c) 2000-2001 Questra. A2B.Platform release 2.2.1 for Win32,
24 July 2001.  Licensed for evaluation only.
    </Copyright>
    <ResponseHeader version="1.0"></ResponseHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <discoverResponse>
      <DiscoverResult>
        <WebServiceProfileList>
          <WebServiceProfile>
            <Description>Remote diagnostics</Description>
            <WebServiceType>RemoteDiagnostics</WebServiceType>
            <WebServiceURL>
          http://ro1-cyoder.ddns.roc.questra.com:8081/a2b/RemoteDiagnostics
            </WebServiceURL>
            <Properties>
              <Property name="MaxDevicePropertyListSize">15</Property>
              <Property name="MaxDeviceCommandListSize">15</Property>
              <Property name="ProxyClassname">
            com.questra.a2b.remotecontrolmanagement.common.RemoteControlProxy
              </Property>
            </Properties>
          </WebServiceProfile>
        </WebServiceProfileList>
      </DiscoverResult>
    </discoverResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Table 12 discover web method sample messages.

The method supports the following parameters:

Parameter	Description
RequestType [in]: RequestType	Identifies the type of request. Valid values are: • DISCOVER_WEB_SERVICES • DISCOVER_PROPERTIES • DISCOVER_COMMANDS • DISCOVER_COMMANDS_DETAIL
Restrictions [in]: RestrictionList	Holds a list of restrictions to filter the query. Valid restrictions are dependent on the request type. See chart below.
Properties [in]: PropertyList	Holds a list of properties to be used by the query. Valid discover properties are dependent on the request type. See chart below.
DiscoverResult [out]: DiscoverResult	The structure containing the results of the discover operation

Table 15 discover web method exception codes.

7.5.1.4 getProperties Operation

This method gets the list of requested properties from the Remote Control web service located on a remote device. A list of property names is passed in and a list of property names and values is returned.

Web Service: Remote Control Management	**Web Method:** getProperties

SOAP Action: http://www.questra.com/a2b/RemoteControlManagement#getProperties

<u>SOAP Request Message</u>:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance/">
  <SOAP-ENV:Header>
    <RequestHeader>
      <Operation>getProperties</Operation>
      <ServiceName>RemoteControlManagement</ServiceName>
      <SecurityContext>
        <Credentials>
          <MemberId>a2btest</MemberId>
          <Password>a2btest</Password>
        </Credentials>
      </SecurityContext>
    </RequestHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <getProperties>
      <MemberId>a2bdevice</MemberId>
      <WebServiceType>RemoteDiagnostics</WebServiceType>
      <PropertyNames>
        <Property name="BOGUS"></Property>
        <Property name="TestProperty1"></Property>
        <Property name="TestProperty2"></Property>
        <Property name="TestProperty3" type="integer"></Property>
        <Property name="TestProperty4" type="string"></Property>
        <Property name="TestProperty5" type="NMTOKEN"></Property>
      </PropertyNames>
    </getProperties>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

<u>SOAP Response Message</u>:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance/">
  <SOAP-ENV:Header>
    <ResponseHeader version="1.0"></ResponseHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <getPropertiesResponse>
      <Properties>
        <Property faultCode="INVALID_PROPERTY_NAME" name="BOGUS">
          Invalid property name.
        </Property>
        <Property name="TestProperty1" type="integer">123</Property>
        <Property name="TestProperty2" type="integer">234</Property>
        <Property name="TestProperty3" type="integer">345</Property>
        <Property name="TestProperty4" type="string">OK</Property>
        <Property name="TestProperty5" type="NMTOKEN">ON</Property>
      </Properties>
    </getPropertiesResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Table 16 getProperties web method sample messages.

The method supports the following parameters:

Parameter	Description
MemberId [in]: MemberId	The member id of the device.
WebServiceType [in]: WebServiceType	The web service type of the web service on the device. Valid values are: • RemoteDiagnostics • RemoteControl
PropertyNames [in]: PropertyList	A list of property names to get from the remote device
Properties [out]: PropertyList	A list of the property values

Table 17 getProperties web method parameters.

The method can raise a *RemoteControlServiceExcepton* exception with the following codes:

Exception Code
REMOTE_CONTROL_MANAGEMENT_GENERAL_ERROR
NOT_AUTHORIZED
INVALID_PARAMETER
INVALID_CONTENT
NO_PROXY_DEFINED
INVALID_PROXY_CLASSNAME

NO_URL_REGISTERED
INVALID_RESPONSE
PROTOCOL_EXCEPTION
IO_EXCEPTION
CONNECT_EXCEPTION

Table 18 getProperties web method exception codes.

7.5.1.5 setProperties Operation

This method sends a list of properties to the Remote Control web service on a remote device, which sets the properties on the remote device. A list of property names and values are passed in, and a list of the properties that could not be set is returned.

Web Service: Remote Control Management	**Web Method:** setProperties
SOAP Action: http://www.questra.com/a2b/RemoteControlManagement#setProperties	

<u>**SOAP Request Message:**</u>

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance/">
  <SOAP-ENV:Header>
    <RequestHeader>
      <Operation>setProperties</Operation>
      <ServiceName>RemoteControlManagement</ServiceName>
      <SecurityContext>
        <Credentials>
          <MemberId>a2btest</MemberId>
          <Password>a2btest</Password>
        </Credentials>
      </SecurityContext>
    </RequestHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <setProperties>
      <MemberId>a2bdevice</MemberId>
      <WebServiceType>RemoteDiagnostics</WebServiceType>
      <Properties>
        <Property name="BOGUS">123</Property>
        <Property name="TestProperty1">5432</Property>
        <Property name="TestProperty2">4321</Property>
        <Property name="TestProperty3" type="integer">6543</Property>
        <Property name="TestProperty4" type="string">TEST</Property>
        <Property name="TestProperty5" type="NMTOKEN">OFF</Property>
      </Properties>
    </setProperties>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

<u>**SOAP Response Message:**</u>

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance/">
  <SOAP-ENV:Header>
    <ResponseHeader version="1.0"></ResponseHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <setPropertiesResponse>
      <Faults>
        <Property faultCode="INVALID_PROPERTY_NAME" name="BOGUS">
          Invalid property name.
        </Property>
      </Faults>
    </setPropertiesResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Table 19 setProperties web method sample messages.

The method supports the following parameters:

Parameter	Description
MemberId [in]: MemberId	The member id of the device.
WebServiceType [in]: WebServiceType	The web service type of the web service on the device. Valid values are: • RemoteDiagnostics • RemoteControl
Properties [in]: PropertyList	A list of property names and values to set on the remote device
Faults [out]: PropertyList	A list of properties that failed to be set on the remote device

Table 20 setProperties web method parameters.

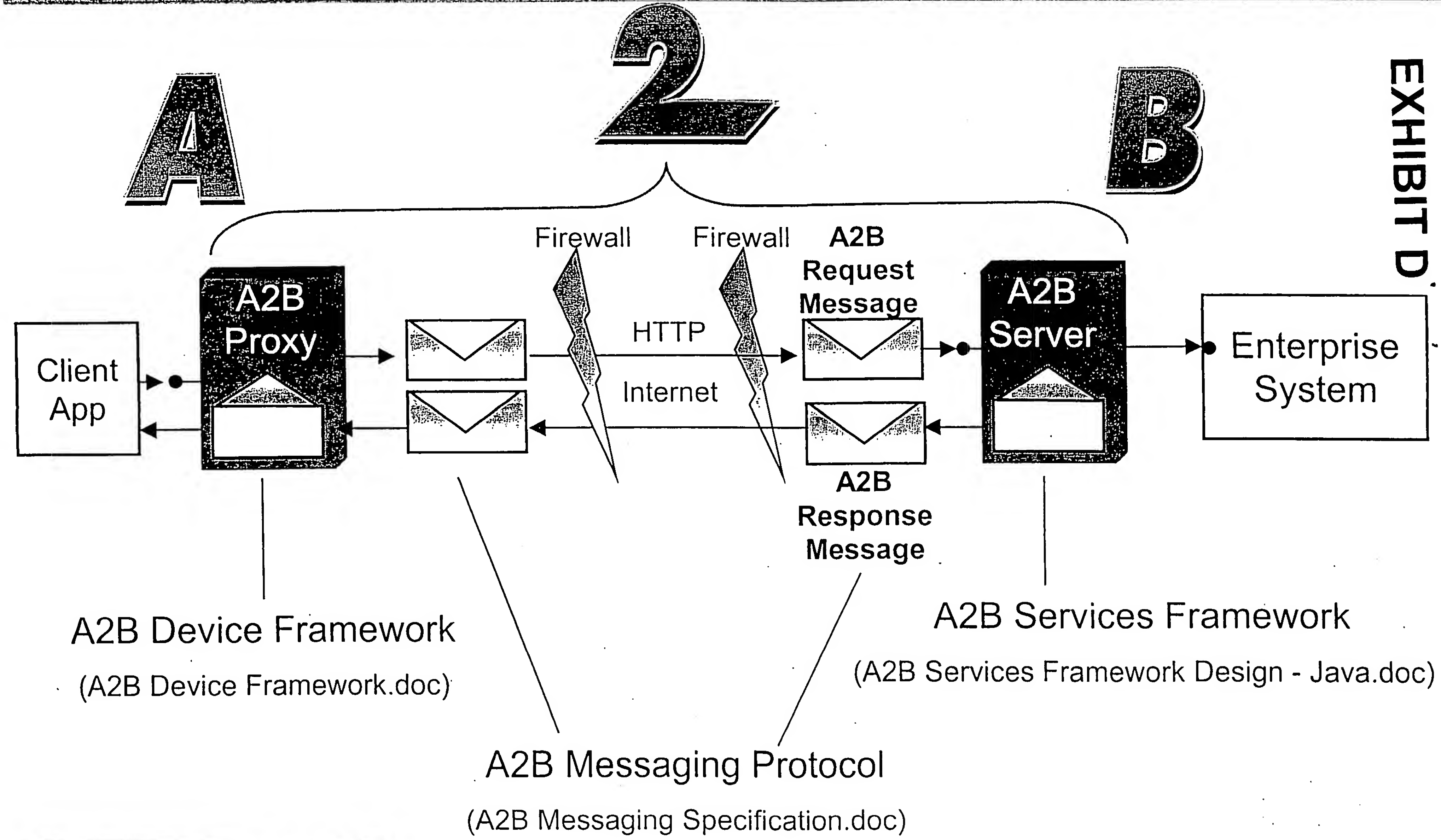
The method can raise a *RemoteControlServiceExcepton* exception with the following codes:

Exception Code
REMOTE_CONTROL_MANAGEMENT_GENERAL_ERROR
NOT_AUTHORIZED
INVALID_PARAMETER
INVALID_CONTENT
NO_PROXY_DEFINED
INVALID_PROXY_CLASSNAME
NO_URL_REGISTERED
INVALID_RESPONSE
PROTOCOL_EXCEPTION

Questra™ CORPORATION

A2B™ Messaging Context Diagram

EXHIBIT D



A2B Device Framework

(A2B Device Framework.doc)

A2B Messaging Protocol

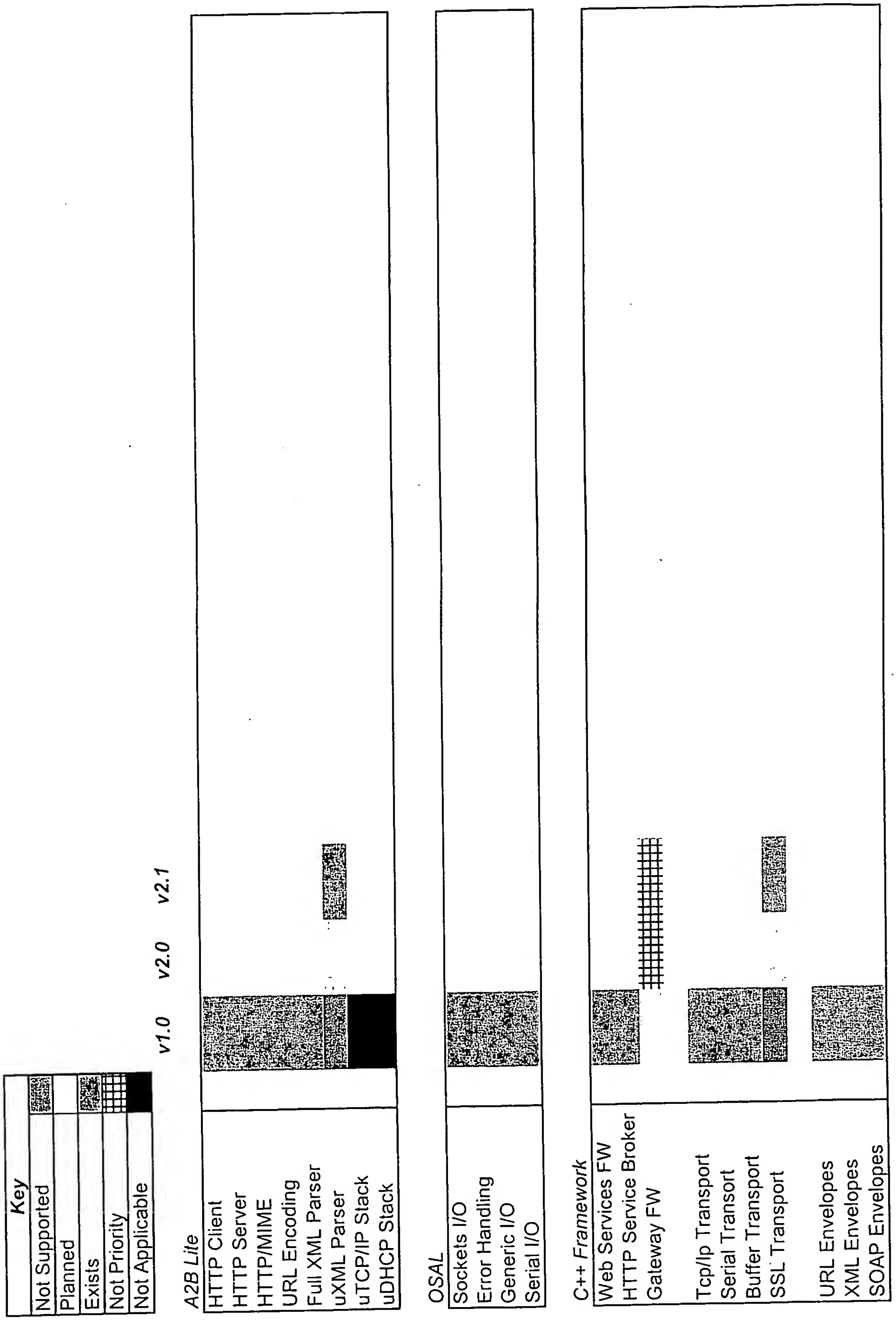
(A2B Messaging Specification.doc)

A2B Services Framework

(A2B Services Framework Design - Java.doc)

[redacted]

EXHIBIT E


Key
Not Supported
Planned
Exists
Not Priority
Not Applicable
v1.0
v2.0
v2.1
A2B Lite
HTTP Client
HTTP Server
HTTP/MIME
URL Encoding
Full XML Parser
uXML Parser
uTCP/IP Stack
uDHCP Stack
OSAL
Sockets I/O
Error Handling
Generic I/O
Serial I/O
C++ Framework
Web Services FW
HTTP Service Broker
Gateway FW
Tcp/Ip Transport
Serial Transort
Buffer Transport
SSL Transport
URL Envelopes
XML Envelopes
SOAP Envelopes